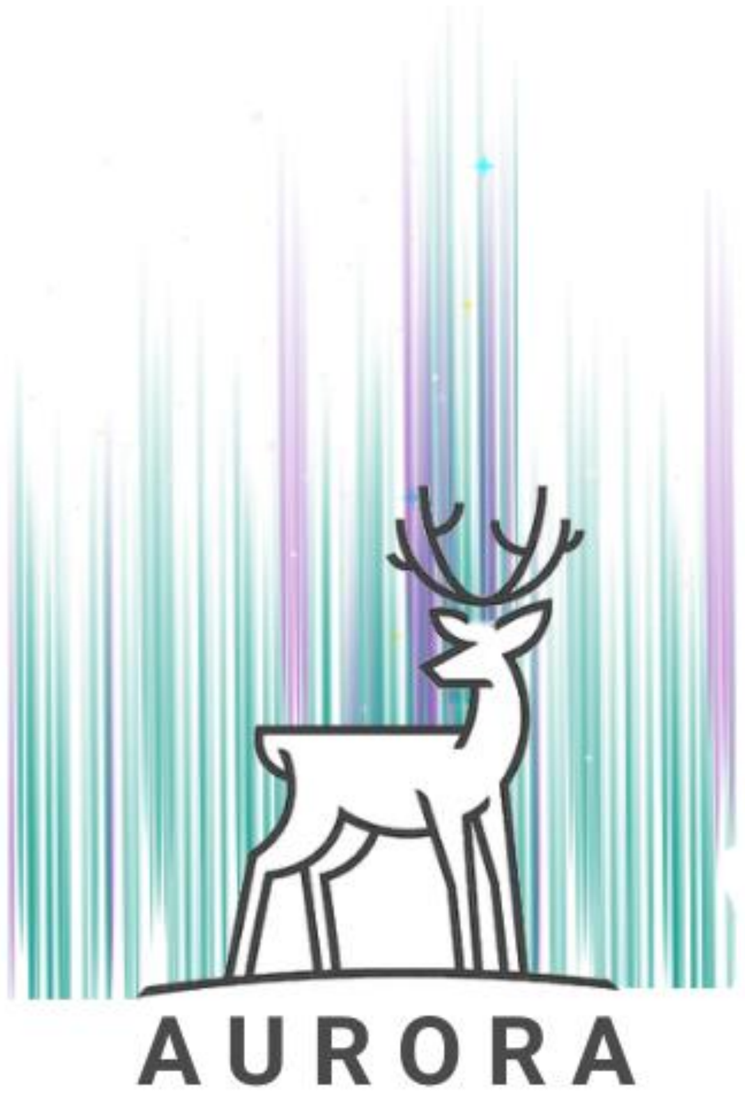
Capstone Project

2024/2025



Group 7

João Capitão | 20221863

Maria Rodrigues | 20221938

Nuno Leandro | 20221861

Vidhi Rajanikante | 20221982

Yehor Malakhov | 20221691

Table of Contents

Data Sources

For Aurora to be able to be more knowledgeable on certain topics, if must be given information. For instance, if PDF files. Due to Aurora's many functionalities, there are various types of PFD files from where Aurora can better her responses and assistance.

Transcript of Academic Records

By analysing the user's academic records, this is, his/hers GPA grades on the previous courses, Aurora can be of better assistance to the user's needs according to his/her knowledge.

Theoretical PowerPoints

One of the ways to better know what the user is studying and needs help with, is by feeding Aurora the PDF files of the respective class power points. As such, Aurora will not to only be able to be more objective on her responses, but also take into account the specific requirements for the specific course, without giving information that the user does not have, nor will learn on the curricular unit.

Practical Exercises File

It is common for a user to have doubts when doing practical exercises for a course, or not being able to correct it. By giving Aurora a photo or a PDF file or the exercises in question, Aurora will be able to assist in these circumstances, making the user achieve a better understanding of his/her errors, and more insightful explanations on why and how each was to resolve the exercise was done.

Relational Database Design

Aurora's database structure consists of fifteen tables, five of which originate from a Many-to-Many relationship between entities. A small description of these tables is represented below, along with. visual representation of the database.

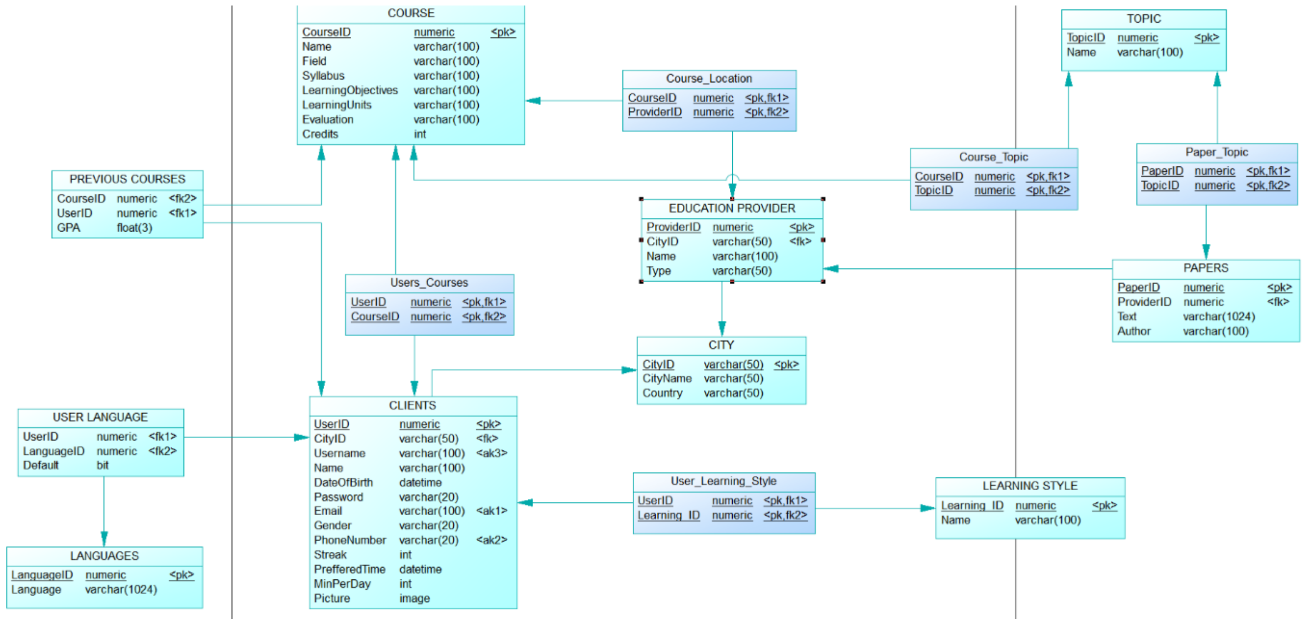


Figure 1 – Aurora's Database Schema (Physical Model)

CLIENTS

The table that stores the user's information is 'CLIENTS'. It can be seen as the main entity in the database. Moreover, only having a single identifier proves a bit vague; as such, this entity has one primary identifier and another three alternative keys. The table's attributes are as follows:

- UserID: primary key, unique identifier for each user.
- CityID: foreign key ('CITY'), indicates the ID of the city where the user lives/is from. Ensures relationship between the 'CLIENTS' and the 'CITY' entities.
- Username: alternative key, alternative unique identifier for each user. Each user can choose their own username, as long as it is unique.
- Name: non-null attribute, represents the name of the user.
- DateOfBirth: non-null attribute, represents the date of birth of each user.
- Password: non-null attribute, represents the user-specified password for each user.
- Email: alternative key, alternative unique identifier for each user. It is the email to which the user's account is connected to.
- Gender: non-null attribute, with three possible field values – *Female*, *Male*, *Non-Binary* and *Prefer Not To Say*.
- PhoneNumber: alternative key, alternative unique identifier for each user. It is the phone number to which the user's account is connected to.
- Streak: non-null attribute, representing the number of consecutive days the user opens/uses Aurora for his/her studies.

- PreferedTime: non-mandatory attribute, referring to the timetable when the user prefers or has the time to make use of Aurora's services.
- MinPerDay: non-null attribute, representing the minimum number of minutes the user wants to learn daily.
- Picture: non-mandatory attribute, consisting of whether or not the user wants to upload a picture to his/her profile.

LEARNING STYLE

'LEARNING STYLE' is a core entity that considers the 'Names' of different learning styles, provided by the 'CLIENTS' to classify and match their preferences to make their experience personalized. Each type of learning style has their own identifier.

COURSES

The 'COURSES' entity can be considered another main entity, given it capability to store information related to all the possible courses one can take. It takes a unique identifier, '*CourseID'*, the '*Name'* of the course, along with its '*Field'*, '*Sylabbus'*, '*LearningObjectives'*, '*LearningUnits'*, '*Evaluation'* metric and guidelines, and '*Credits'*.

PREVIOUS COURSE

'PREVIOUS COURSES' is an entity necessary to establish a personal connection between the COURSES and CLIENTS, this is, the courses that the user has already taken. Thus, it stores the '*GPA'* grade of the user on the respective courses. It does not have any unique identifiers, having only as foreign keys the primary keys from both the tables it is connected to: '*CourseID'* and '*UserID'*.

USER_COURSES

This table, 'USER_COURSES', originates from the Many-To-Many relationship between CLIENTS and COURSES. It stores the courses the user is currently taking, keeping as its only attributes the primary keys from both the tables it is connected to: '*CourseID'* and '*UserID'*. Both these attributes are primary keys and foreign keys.

LANGUAGES

'LANGUAGES' maintains a list of all the languages supported by the chatbot to normalize the language communication amongst users. It has a primary key, 'LanguageID', and the mandatory 'Name' of the language.

USER LANGUAGE

This entity shares a Many-to-One relationship with 'LANGUAGES' and 'CLIENTS'. It associates the user's preferred language to normalize language data across the users and facilitate the explanation purposes of the chatbot. It only consists of foreign keys because it solely associates the users with one or more languages that they prefer to use when using the chatbot. It also has the 'Default' attribute as a bit to set the preferred language to default or not as the user may juggle between different languages for further explanation.

USER_LEARNING_STYLE

The 'USER_LEARNING_STYLE' is a linking entity formed by a Many-to-Many connection between 'CLIENTS' and 'LEARNING_STYLE' to enable customized educational experiences.

EDUCATION PROVIDER

The 'EDUCATION PROVIDER' table main purpose is to store the different entities that provide any sort of course or education. It has as attributes a unique identifier, '*ProviderID'*, the '*Name'* of the provider (i.e. NOVA IMS) and the '*Type'* of education it provides (i.e. online or in-person). Due to its connection to CITY, it also has the foreign key of the information on the '*CityID'* the education provider is located, if applicable. In addition, it is necessary to relate the user educational needs presented to the chat and the user location with the courses and articles that will be provided to the user that are stored in the database.

COURSE_LOCATION

Similar to USER_COURSES, this entity results from a Many-To-Many relationship between 'EDUCATION PROVIDER' and 'COURSE'. It only takes as attributes the primary keys from the tables it originates from, making those also foreign keys.

CITY

A core entity that stores the geographical location of the 'CLIENTS' and the 'EDUCATION PROVIDER', simply having attributes such as 'CityName' and 'Country', as well as a primary key 'CityID'. It is essential to know about the origin or current geolocation of the 'client' and its 'provider of education' to simplify prompt delivery.

PAPERS

A core entity storing the research papers and academic resources associating itself with the 'PAPER_TOPIC' and 'EDUCATION_PROVIDER' entities to help with the citation and confidentiality of the resource. This table has as attributes 'ProviderID' – foreign keys from 'EDUCATION PROVIDER' – 'Author', and 'Text' which is information retrieval from the paper in question. This entity is vital, since it will contain the information needed to answer the questions posed by the user.

TOPIC

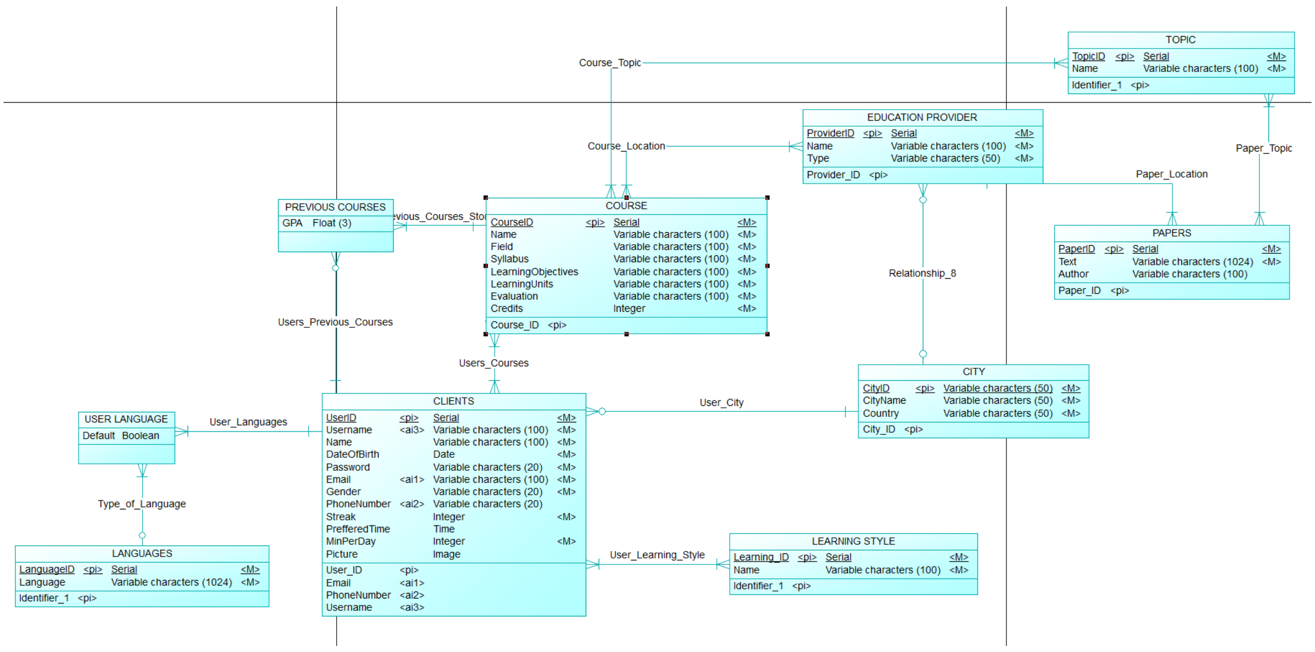
The table 'TOPIC' contains the numerous topics that categorize the papers or courses, thus facilitating filtering and organizing study content. It has an identity attribute, 'TopicID' and 'Name' where the topic is specified.

COURSE_TOPIC

Connecting the 'COURSE' and 'TOPIC' entities, this linking entity connects the courses to their respective topics, thus simplifying topic-based course searches done by the model. As such, the table 'COURSE_TOPIC' only has two attributes: 'CourseID' and 'TopicID', both primary keys from the entities previously mentioned.

PAPER_TOPIC

'PAPER_TOPIC' is a similar to the 'COURSE_TOPIC' linking entity, with the difference of just connecting the 'TOPIC' entity to the 'PAPERS' entity, making the organization and retrieval of information easier for prompt queries from 'CLIENTS'. Its attributes are 'PaperID' and 'TopicID', and both are primary keys.



TOPIC
TopicID <pi> Serial <M>
Name Variable characters (100) <M>
Identifier_1 <pi>
Course_Topic
EDUCATION PROVIDER
ProviderID <pi> Serial <M>
Name Variable characters (100) <M>
Type Variable characters (50) <M>
Provider_ID <pi>
Course_Location
Paper_Topic
Paper_Location
PREVIOUS COURSES
GPA Float (3)
vious_Courses_Stor
COURSE
CourseID <pi> Serial <M>
Name Variable characters (100) <M>
Field Variable characters (100) <M>
Syllabus Variable characters (100) <M>
LearningObjectives Variable characters (100) <M>
LearningUnits Variable characters (100) <M>
Evaluation Variable characters (100) <M>
Credits Integer <M>
Course_ID <pi>
PAPERS
PaperID <pi> Serial <M>
Text Variable characters (1024) <M>
Author Variable characters (100)
Paper_ID <pi>
Relationship_8
Users_Previous_Courses
Users_Courses
CITY
CityID <pi> Variable characters (50) <M>
CityName Variable characters (50) <M>
Country Variable characters (50) <M>
City_ID <pi>
User_City
USER LANGUAGE
Default Boolean
User_Languages
CLIENTS
UserID <pi> Serial <M>
Username <ai3> Variable characters (100) <M>
Name Variable characters (100) <M>
DateOfBirth Date <M>
Password Variable characters (20) <M>
Email <ai1> Variable characters (100) <M>
Gender Variable characters (20) <M>
PhoneNumber <ai2> Variable characters (20)
Streak Integer <M>
PrefferedTime Time
MinPerDay Integer <M>
Picture Image
User_ID <pi>
Email <ai1>
PhoneNumber <ai2>
Username <ai3>
Type_of_Language
LANGUAGES
LanguageID <pi> Serial <M>
Language Variable characters (1024) <M>
Identifier_1 <pi>
User_Learning_Style
LEARNING STYLE
Learning_ID <pi> Serial <M>
Name Variable characters (100) <M>
Identifier_1 <pi>

The Conceptual model for Aurora's database is as follows:

Figure 2 - Aurora's Database Schema (Conceptual Model)

Chatbot Architecture

In this section, we will delve further into Aurora's many capabilities. As a chatbot, Aurora will be able to respond to company queries, database modification, RAG integration, and handling user intents.

Each of Aurora's features will be represented as a user story, which will include who the user is, what they want to achieve, and why the feature is important. Moreover, each user intention will be accompanied by a visual representation of the data flow, emphasizing the user's functionalities.

Our Chatbot Intentions:

1. Saving Education from PDF

- **User Story:**

  As a user,

  I want to be able to extract and save educational content from my uploaded PDFs,

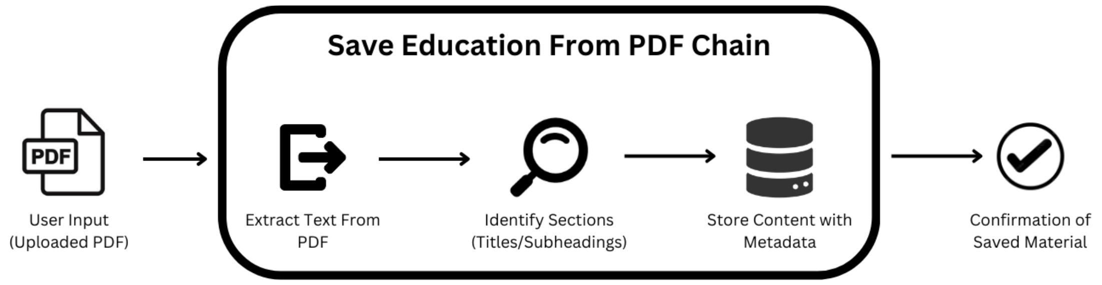
  So that I can easily access and organize the material for future use. I also want to ask for explanations, summaries, and context directly from the study material my professor gave me. This ensures I focus on the course-specific terms and concepts, avoiding confusion caused by unrelated materials or alternative approaches.

- **Goal:**

  Extract and save educational content from user-uploaded PDFs for future use.

- **Implementation**:

  First, a document loader is used to extract text from the PDF. The extracted content is processed to identify sections (e.g., chapter titles, subheadings). This information is then stored in a structured format (database) with metadata like subject and title. Finally, the system returns a confirmation message that the material has been successfully saved.


Save Education From PDF Chain
PDF
User Input (Uploaded PDF)
Extract Text From PDF
Identify Sections (Titles/Subheadings)
Store Content with Metadata
Confirmation of Saved Material

2. Creating Quizzes from Used Material

- **User Story:**

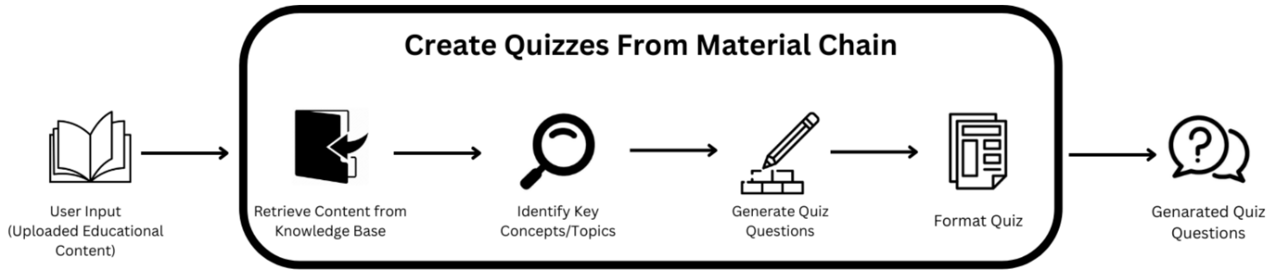
  As a user,

  I want to be able to generate quizzes from my uploaded educational material because my professor didn't give access to past exams or exercises on my Introduction to Biology course,

  So that I can test my understanding, feel more confident about my knowledge, and to help me identify weaknesses in my understanding of the subject.

- **Goal:**
  To generate quizzes based on uploaded educational material.

- **Implementation:**
  Based on the user-upload educational content, pertinent content stored in the knowledge base will be retrieved. Key concepts, terms and topics will be identified by a language model and used to generate quiz questions (e.g., multiple choice or true/false questions) with a question generation tool. Finaly the quiz will be formatted in a user-friendly string to ensure readability and returned to the user.


Create Quizzes From Material Chain
User Input (Uploaded Educational Content)
Retrieve Content from Knowledge Base
Identify Key Concepts/Topics
Generate Quiz Questions
Format Quiz
Genarated Quiz Questions

3. Creating Flashcards from Memory

- **User Story:**

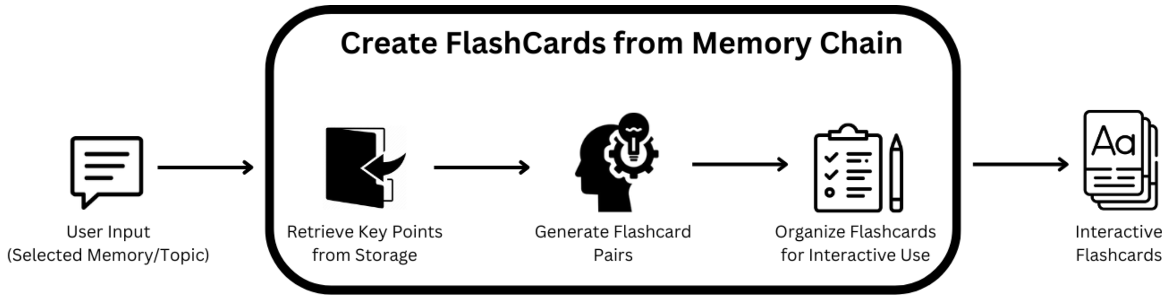
  As a user,

  I want to create flashcards based on my previously learned material, because I am a visual learner and quizzes feel tedious.

  So that, I can efficiently review and retain knowledge on the key concepts of the topic in a easy-going and fun way.

- **Goal:**
  Create flashcards for the user to review the subject he/she's studying based on previously learned material or stored memory.

- **Implementation:**
  By considering past learned material by the user, the retrieval of key points and terms using a vector database or structured storage will be imputed to a chain to generate flashcard pairs (e.g., term/definition or question/answer). In the end the flashcards are organized and presented to the user in a suitable format that ensures interactivity and readability, so that initially only one side of each flashcard pair is visible to the user at a time.


Create FlashCards from Memory Chain
User Input (Selected Memory/Topic)
Retrieve Key Points from Storage
Generate Flashcard Pairs
Organize Flashcards for Interactive Use
Interactive Flashcards
Aa

4. Retrieve Papers From SQL Database

- **User Story:**

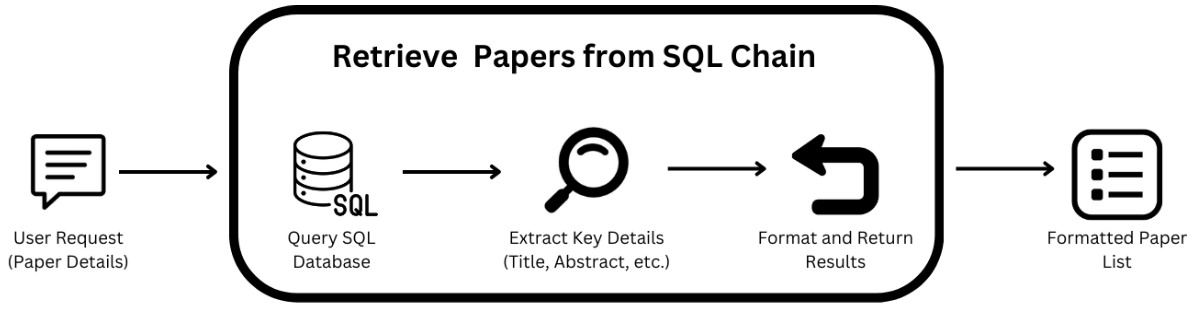
  As a user,

  I want to be able to find and view relevant academic papers or articles from the database based on the subject that Aurora is helping me to learn,

  Because it will help me find more resources and different perspectives on the subject to help me build a solid understanding of the topic.

- **Goal:**
  Retrieve and display relevant academic papers or articles from an SQL database.

- **Implementation:**
  Based on the user input, a chain that queries the SQL database in search for relevant academic papers or articles will be implemented. The retrieved documents will be processed to extract titles, abstracts, and other relevant details. Finally, the results will be structured and displayed in a user-friendly format.


Retrieve Papers from SQL Chain
User Request (Paper Details)
Query SQL Database
SQL
Extract Key Details (Title, Abstract, etc.)
Format and Return Results
Formatted Paper List

5. Update User Info from the Chat

- **User Story:**

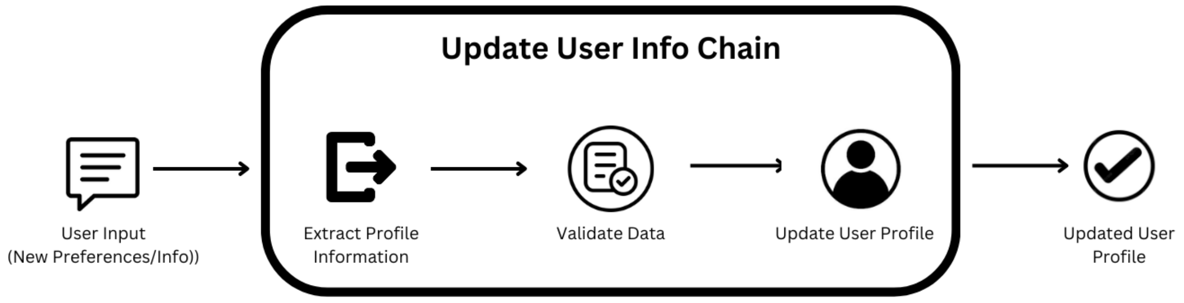
  As a user,

  I want to have a truly customizable learning experience where the process is made for me and there's an easy way to update my needs and aspirations with the learning experience,

  So that Aurora will adapt the learning pace and recommendations to make me feel more involved and captivated in my study.

- **Goal:**
  Update the user's profile or preferences based on chat interactions.

- **Implementation:**
  Updating the user's profile and learning preferences will be made via a chat interaction. This is made possible by implementing a conversational chain to analyse user inputs for profile-related information (e.g., preferred subjects, learning goals). The extracted information is validated and finally updated in the user's profile database.



6. Summarize from Any Files

- **User Story:**

  As a user,

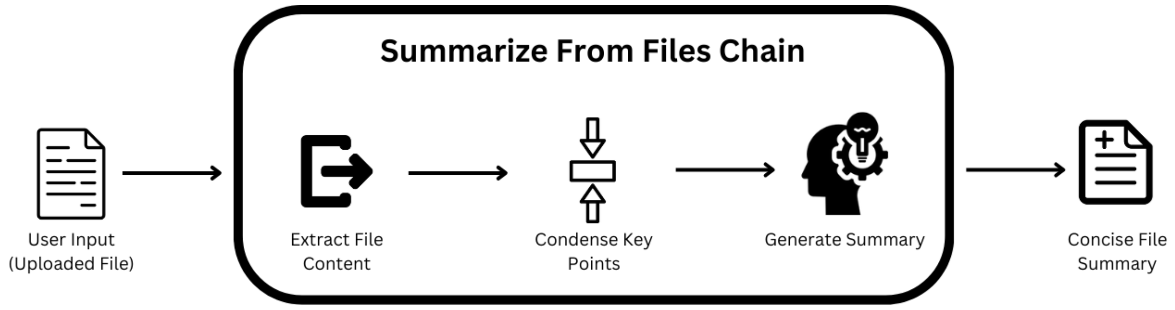
  I want to be able to receive concise summaries of any files I upload,

  So that I can quickly understand the key points and concepts without having to read the entire document, saving me time.

- **Goal:**
  Provide concise summaries of user-uploaded files (e.g., PDFs, Word documents).

- **Implementation:**

  First, a document loader is used to extract content from various file types. Then a summarization tool is used to condense the extracted text into key points or a short overview and returned in a user-friendly format.



7. Citation of Provided Report

- **User Story:**

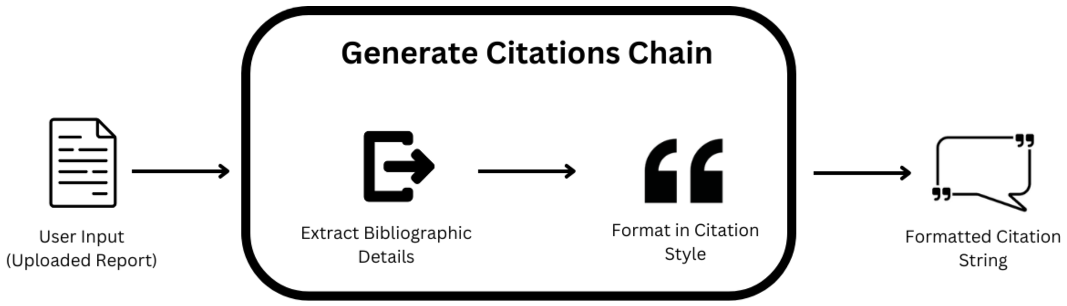
  As a user,

  I want to be able to generate formatted citations from the documents I provide without having to manually search for this information,

  So that I can include accurate references in my work, with less effort and time spent.

- **Goal:**
  Generate formatted citations for academic or professional reports.

- **Implementation:**
  Based on user-provided text or documents, the bibliographic details (e.g., author, title, publication date) are extracted and formatted into the desired citation style (e.g., APA).
  After this, a formatted citation is returned to the user.



Generate Citations Chain
User Input (Uploaded Report)
Extract Bibliographic Details
Format in Citation Style
Formatted Citation String

8. Motivational Support

- **User Story:**

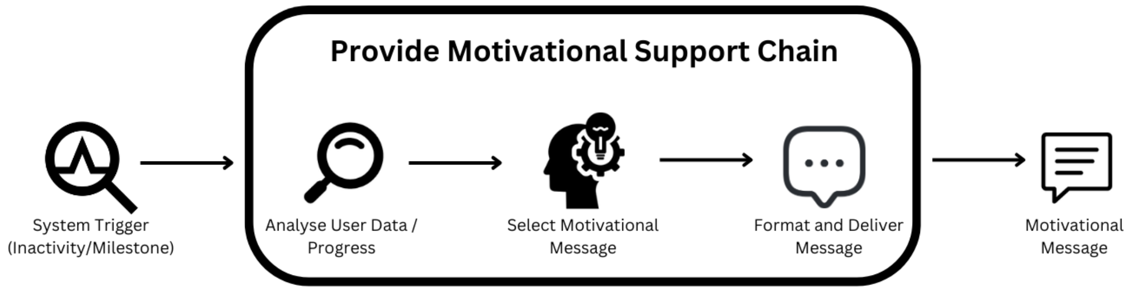
  As a user,

  I want to receive motivational messages based on my progress, milestones, or difficulties,

  So that I feel encouraged to continue studying, even when I struggle or feel demotivated.

- **Goal:**
  Provide motivational messages to encourage and engage users during their learning journey.

- **Implementation:**
  First the user input or his/her progress data is analysed to determine the appropriate motivational response. A chain is triggered to generate or retrieve motivational messages and deliver them as text.


Provide Motivational Support Chain
System Trigger (Inactivity/Milestone)
Analyse User Data / Progress
Select Motivational Message
Format and Deliver Message
Motivational Message

9. Application Sends

- **User Story:**

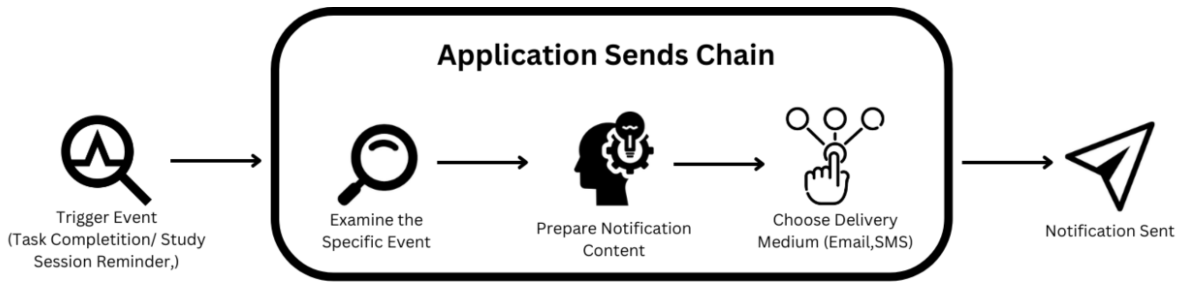
  As a user,

  I want to receive notifications or updates via email,

  So that I can stay informed about my tasks and progress or be reminded of my study time with Aurora. As I often forget of my tasks, and I am struggling to develop a daily study habit, this functionality is a much-needed help.

- **Goal:**
  Send notifications, reminders, or other updates to the user via a messaging platform.

- **Implementation:**
  When certain events occur, such as task completion or upcoming deadlines, a chain is triggered to create and send notifications in the form of task reminders, quiz links, or progress updates.


Application Sends Chain
Trigger Event (Task Completition/ Study Session Reminder,)
Examine the Specific Event
Prepare Notification Content
Choose Delivery Medium (Email,SMS)
Notification Sent

10. Content Categorization and Tagging

- **User Story:**

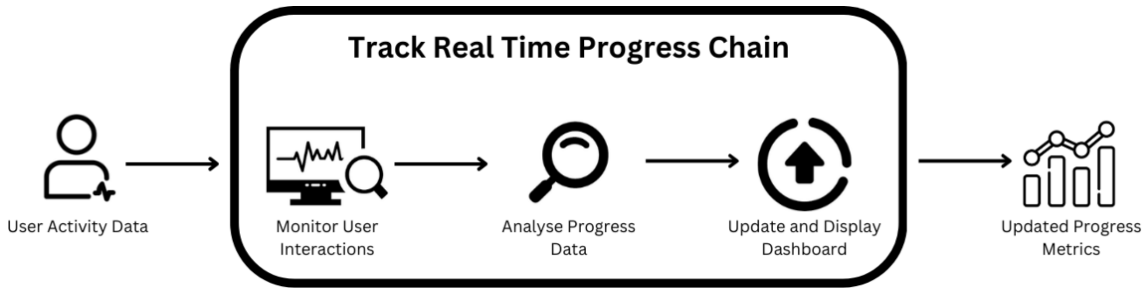
  As a user,

  I want to easily retrieve and organize the material that I present Aurora,

  Because as a disorganized and impatient person, I will be able to know in no time what materials I have presented Aurora that I can use, helping me organize my study and how many times I need to cover all the materials.

- **Goal:**
  Categorize and tag uploaded content for easier retrieval and organization.

- **Implementation:**
  When the user uploads documents, the main topics, keywords and difficulty level are identified to help a classification model to assign tags. In the end, the categorized content and metadata are saved in the database.


Track Real Time Progress Chain
User Activity Data
Monitor User Interactions
Analyse Progress Data
Update and Display Dashboard
Updated Progress Metrics

11. Real-Time Progress Tracking

- **User Story:**

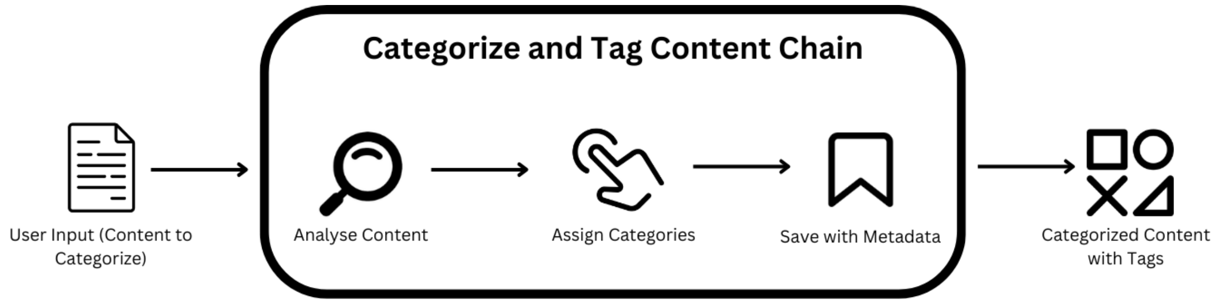
  As a user,

  I want to see real-time updates on my learning progress,

  So that I can keep track of my progress with Aurora, feel rewarded and motivated by seeing my daily streak and statistics of my time with Aurora or to understand that my study habits still need improvement.

- **Goal:**
  Track and display the user's learning progress in real-time.

- **Implementation:**
  The user interactions with the chat (e.g., completed quizzes, reviewed flashcards) are monitored and generate progress data that is stored in the database. This data will be used to calculate metrics (e.g., completion rates, accuracy) and used to generate visual progress updates or insights for the user, such as charts, graphs, or progress bars.


Categorize and Tag Content Chain
User Input (Content to Categorize)
Analyse Content
Assign Categories
Save with Metadata
Categorized Content with Tags